CityScope: Real Estate & Community Data Explorer

**Course:** CSIS 4495-071 - Applied Research Project - Fall 2025  
**Student:** Sahan Pattinikutige Nonis - 300389470 (Team Lead)  
**Instructor:** Dr. Bambang Sarif

# Introduction

The rising cost of living has made housing decisions increasingly complex, especially for students, families, and individuals relocating to new cities. While real estate websites provide listings, they often fail to integrate key contextual data such as school quality, transit accessibility, and neighborhood affordability. CityScope addresses this gap by creating a web application that leverages data analytics to provide a holistic view of neighborhoods. Rather than simply displaying listings, the system highlights patterns and tradeoffs to help users make informed housing choices.

**Problem Statement**Urban residents often struggle to evaluate neighborhoods due to fragmented and non-transparent information on housing affordability, school quality, and transportation access. This makes decision-making about where to live time-consuming and uncertain.

**Knowledge Gap**While real estate platforms provide listings, they rarely integrate multiple datasets (housing, schools, transit) into a single, data-driven view. There is also limited focus on user-friendly analytics that allow comparison across neighborhoods beyond price alone.

**Hypothesis**By integrating diverse urban datasets into a centralized web platform and applying analytics, users will be able to make more informed decisions about neighborhood suitability, improving transparency and reducing time spent searching.

**Contribution to Knowledge**CityScope contributes by demonstrating how open data and visualization techniques can be combined into a practical decision-support tool. This project highlights the value of applying data analytics in urban studies, creating a replicable model for combining public data sources into actionable insights.

# Proposed Research Project

Existing platforms primarily focus on property listings and basic filters such as price and location. Users are left to manually compare schools, commute times, and affordability across multiple sources. This fragmented process makes it difficult to assess the true value of a neighborhood, particularly for those unfamiliar with the area. CityScope seeks to solve this problem by combining housing, school, and transit data into a single interface with meaningful analytics that support awareness and decision-making.

### **Objectives**

The primary objective of CityScope is to design and implement a web-based system that provides users with accessible, data-driven insights into housing and neighborhoods. Specific goals include integrating housing prices with external factors, creating visualizations of price trends and affordability, and introducing a composite neighborhood score that balances cost, accessibility, and education quality. The system will demonstrate how analytics can enhance everyday decision-making in real estate contexts.

### **Research Design & Methodology**

CityScope will be developed using a monorepo structure with NestJS as the backend framework and Next.js for the frontend interface. Housing, school, and transit datasets will be sourced from open data repositories such as municipal databases and Kaggle. The backend will preprocess these datasets to calculate key metrics, including average rent per neighborhood, historical price trends, and transit accessibility scores. A composite “neighborhood score” will be generated using a weighted formula that combines affordability, education, and transportation. The frontend will present this information through interactive dashboards and charts, enabling users to filter results and compare neighborhoods.

The research design for CityScope follows an applied, data-driven development approach, integrating urban datasets into a functional decision-support platform. The methodology is structured in four phases:

**1. Data Acquisition**

* Identify and collect relevant open datasets (e.g., housing prices, rental listings, school ratings, public transit routes, and crime statistics).
* Ensure datasets are reliable, updated, and accessible via open APIs or government portals.
* Apply data cleaning techniques to handle missing values, inconsistencies, and formatting issues.

**2. Data Integration and Processing**

* Normalize heterogeneous datasets into a unified schema.
* Use preprocessing methods (e.g., scaling, encoding categorical data) to prepare data for visualization and analytics.
* Store data in a lightweight relational database (e.g., SQLite or PostgreSQL) for structured querying.

**3. Data Analytics and Visualization**

* Apply descriptive analytics to calculate key metrics such as average housing cost, school performance index, and commute accessibility.
* Design comparative metrics that allow neighborhoods to be ranked or filtered according to user-selected criteria.
* Develop visualization modules (heatmaps, charts, and comparative dashboards) to present results clearly.

**4. Platform Development**

* Build a web-based application using a modern stack (React/Next.js for frontend, Node.js/Express for backend).
* Implement interactive features that let users filter neighborhoods based on preferences (budget, school quality, commute).
* Ensure responsiveness and usability across devices through iterative user interface testing.

**Validation Strategy**

* Conduct case studies on selected neighborhoods to test the accuracy of integrated analytics.
* Perform small-scale user testing to gather feedback on usability and decision-making support.

### **Technology Stack**

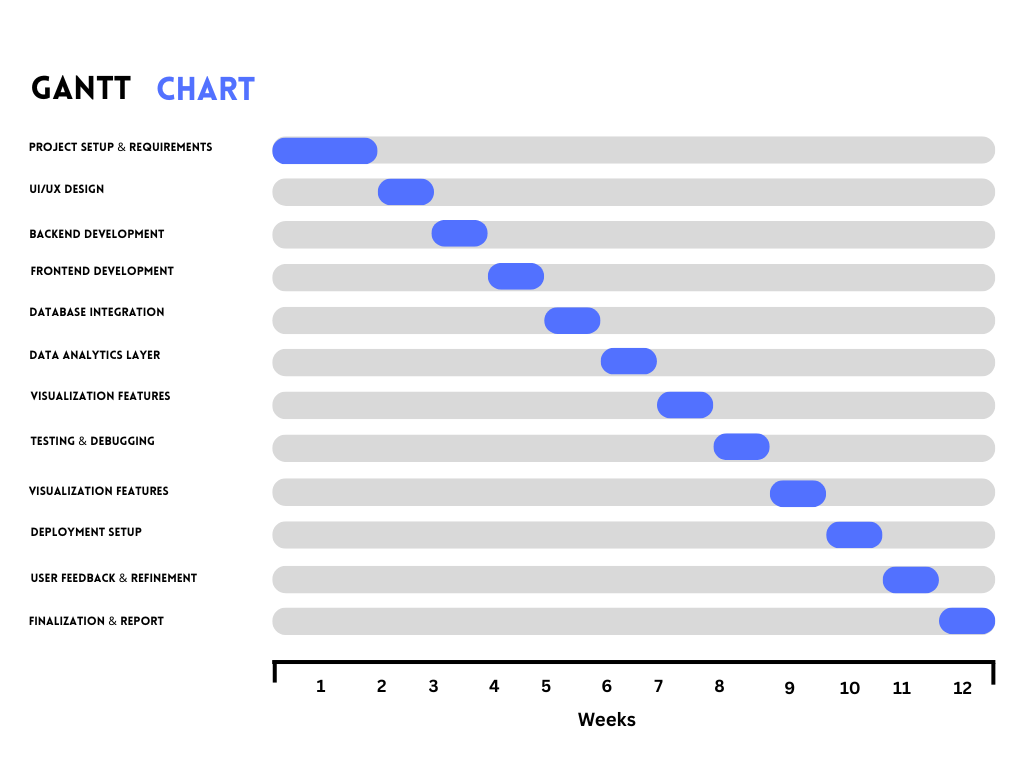
* **Platform:** Docker
* **Frontend:** Next.js (React) with Chart.js/Recharts for visualizations
* **Backend:** NestJS (Node.js) with REST APIs
* **Database:** PostgreSQL (via Prisma ORM).
* **Deployment:** Vercel (frontend) + Railway/Render/Fly.io (API & Postgres)

# Expected Results

The expected outcome is a functional prototype that demonstrates how integrated analytics can guide housing decisions. The system will present property information alongside visual insights such as line charts for price trends and scatterplots for commute versus cost tradeoffs. Users will be able to compare neighborhoods not only by price but also by overall quality as measured by the composite score. The project will showcase the value of analytics in personal decision-making and provide a foundation for future expansion into a full-featured relocation tool.

# Project Planning and Timeline

| **Week** | **Milestone** | **Deliverables** |
| --- | --- | --- |
| 1 | Project Setup & Requirements | Project repo initialized, finalized requirements document |
| 2 | UI/UX Design | Wireframes, mockups of main pages (search, results, dashboard) |
| 3–4 | Backend Development | NestJS APIs for housing data ingestion, processing, and scoring |
| 5 | Frontend Development | Next.js components for search, filters, and basic charts |
| 6 | Database Integration | PostgreSQL schema set up, connected via Prisma ORM |
| 7 | Data Analytics Layer | Neighborhood scoring, trend analysis, initial visualization logic |
| 8 | Visualization Features | Charts, maps, and interactive comparisons implemented |
| 9 | Testing & Debugging | Unit tests, integration tests, bug fixes |
| 10 | Deployment Setup | Deployment to Vercel + Railway/Render with live demo |
| 11 | User Feedback & Refinement | Peer/instructor feedback, improvements to UI and analytics |
| 12 | Finalization & Report | Complete working system, final report, and presentation slides |



# Project Contract

I, Sahan Pattinikuttige Nonis, commit to completing the backend, frontend, research design, and analysis independently. Work logs will be updated weekly to track hours and progress.

# Work Log

| **Student Name** | **Date** | **Hours** | **Work Done** |
| --- | --- | --- | --- |
| Sahan Nonis | Sept 17, 2025 | 6 | Drafted proposal introduction & problem statement |

# References

**National Housing Data** – Government Open Data Portals.

**Kaggle Datasets** – Housing Prices, School Ratings, Transit Accessibility.

**Recharts Documentation** (https://recharts.org) – Visualization library for React.

**Next.js Documentation** (https://nextjs.org) – React framework for frontend development.

**NestJS Documentation** (https://nestjs.com) – Backend framework for API development.